ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

Movie Recommendation System based on Deep Learning

Chaoju Hu, Beichen Zhao

School of Control and Computer Engineering, North China Electric Power University, Baoding 071000, China.

Abstract

In the current traditional collaborative filtering recommendation algorithm, there is a lack of mining potential features in the data, and the impact of the distance between the scoring time and the recommendation time on the recommendation is not considered. To solve this problem, this paper proposes a movie recommendation model based on deep learning and time factor. The experimental results in the MovieLens 1M data set show that after the introduction of deep learning and time factors, the MAE value is reduced by about 0.02-0.04, and the accuracy of the recommendation is improved.

Keywords

Movies; Collaborative Filtering Recommendation; Convolutional Neural Network; Word2Vec; Time Factor.

1. Introduction

Nowadays, with the continuous development of the Internet industry, more and more data are spewing out. How to find useful data for users from the massive data and recommend them to users has become a problem that needs to be solved. Movie recommendation is a more important aspect of the recommendation system. In the recommendation process, it is necessary to meet the user's interest needs and recommend movies with higher ratings to users. Traditional recommendation methods include collaborative filtering recommendation, content-based recommendation and hybrid recommendation. The content-based recommendation method was the first to be used. The idea of the content-based recommendation method is very simple, which is to recommend items that are similar to their past interests to users. The disadvantage is that it relies too much on the attribute characteristics marked by the item, requires high marking characteristics, and the probability of recommendation failure is high. Recommendation based on collaborative filtering was first proposed in 1992. The collaborative filtering recommendation algorithm includes user-based collaborative filtering recommendation and item-based collaborative filtering recommendation. User-based collaborative filtering recommendation calculates the similarity between users according to user preferences, and finds out The K most similar users make recommendations. Item-based collaborative filtering recommendation is to calculate the similarity between items to make recommendations. Hybrid recommendation is a combination of multiple recommendation methods to improve the performance of the recommendation system. At present, the most widely used algorithm is collaborative filtering. The most classic method of collaborative filtering recommendation is matrix factorization. Matrix factorization can use the interactive information between users and movies to make recommendations for users, but general matrices have sparseness and cold start problems, It has caused a great challenge to the research of recommender system. In recent years, people have incorporated deep learning into the research of recommender systems. Huang Liwei and others analyzed the development status and future development trends of deep learning in recommendation systems in [6]. Deep learning can reflect the characteristics of data from a deeper level. Sentiment analysis has been a hot research direction in deep learning in recent years. It shows the position of this sentence by digging out the most characteristic emotional words in many texts.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

As in the movies rated by users, the collaborative filtering recommendation method can only simply calculate a score between 1 and 5, and the scoring effect is relatively rough. Therefore, this paper uses the method of Word2Vec+CNN to extract the features of movie names, and constructs a fusion matrix of user features and movie features in addition to the existing user-movie rating matrix. After obtaining the user-movie-rating matrix, it is also necessary to consider the changes in the user's interest caused by the passage of time.

2. Related work

2.1 User-based collaborative filtering recommendation algorithm

The recommendation algorithm is an algorithm that predicts items that users may like through a series of mathematical calculations. Among them, collaborative filtering recommendation algorithm is the most widely used one. The user-based collaborative filtering recommendation algorithm mainly considers the similarity between the user and the user. As long as it finds the items that similar users like, and predicts the target user's rating of the corresponding item, several items with the highest ratings can be found and recommended to the user. W. Zhou et al. weighted the number of common scoring items in [1], which improved the shortcomings of the Person correlation coefficient and effectively improved the recommendation efficiency. In [2], Y. Zeng et al. proposed a method based on user characteristics and user ratings for the cold start problem in collaborative filtering recommendation algorithm. The recommendation algorithm is optimized according to the user's basic attributes and the user's historical scoring record, so that the recommendation efficiency is improved. R.Ji et al. proposed a collaborative filtering recommendation algorithm based on user characteristics in [3]. By analyzing user characteristics, then introducing user preference and trust to improve the accuracy of recommendation. In [10], M. Jianjun et al. considered the user's behavior characteristics into the recommendation algorithm, and used the collaborative filtering algorithm to find out the user's common behavior characteristics during the visit, which effectively solved the cold start problem in the traditional recommendation algorithm. Based on the traditional recommendation algorithm and the personalized recommendation algorithm, this paper integrates the neural network to construct a deep recommendation model, so that the features in the user-movie data can be extracted more effectively.

2.2 Word2Vec

Word2Vec is a word embedding model. In the traditional one-hot encoding, only words are symbolized, the words that appear are marked as 1, and the rest are represented by 0. The disadvantage of one-hot encoding is that the vector dimensionality is too high, and the meaning of words cannot be given. The Word2Vec model solves the above two problems. Word2Vec mainly includes two models, CBOW and skip-gram. The CBOW model predicts the target word based on the context to obtain the word vector, and skip-gram predicts the surrounding words based on the target word. To train to get the word vector. The model diagram is shown in Figure 1.

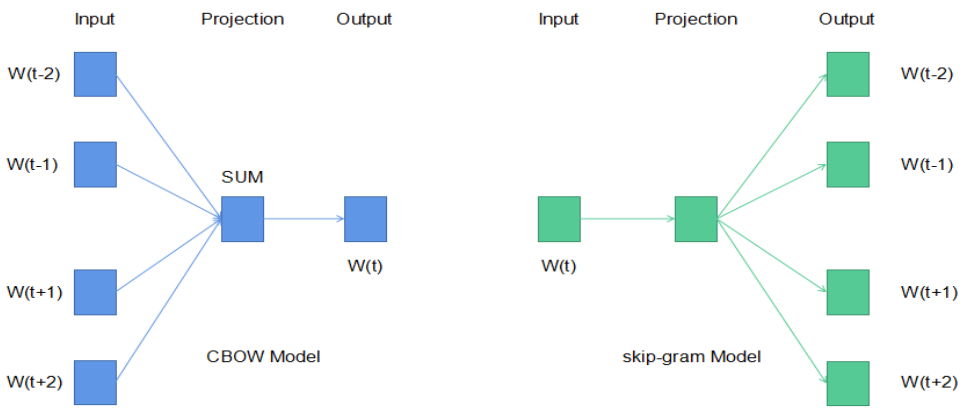

Figure 1. CBOW and skip-gram model

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

Mikolov et al. first proposed Word2Vec in [5]. The article mainly demonstrates the efficiency of word2vec from the time complexity, but there are relatively few descriptions of the principles and specific training methods. Y. Xiao et al. proposed a hybrid recommendation algorithm based on collaborative filtering and Word2Vec in [4], using the Word2Vec model to train the label information of mobile data to obtain the similarity between the labels, and recommend applications to users based on the similarity. Finally, according to the user's feedback behavior, the recommendation result is mixed with the weight.

2.3 CNN

Convolutional Neural Networks (CNN) have been applied to natural language processing in recent years. Convolutional neural network is composed of input layer, convolution layer, pooling layer, fully connected layer and output layer. First, the short text is put into a fixed-size two-dimensional matrix, and then a suitable convolution kernel is set to perform the convolution operation, and the feature mapping matrix is obtained by setting the corresponding convolution kernel and multiplying the dot product of the input matrix. The obtained feature mapping matrix is subjected to a maximum pooling operation, and then concatenated with all the scalars obtained by convolution to form a final one-dimensional vector. Kim et al. [7] first proposed the application of convolutional neural networks to sentence classification. The training model is shown in Figure 2. It turns out that unsupervised word vector preprocessing training is an important part of deep learning in natural language processing

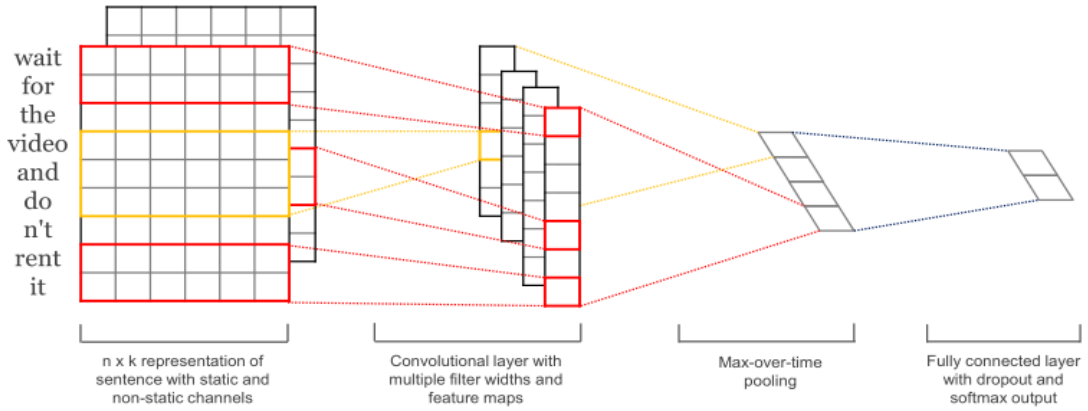

Figure 2. Text convolutional neural network model

H. Zarzour et al. [8] proposed a review classification method based on convolutional neural network, which extracts review features through convolutional neural network and predicts whether the review is positive or negative. Based on these additional information, the recommendation system can be enhanced. Interpretability. Y. Zhang et al. [9] believe that traditional recommendation methods ignore the feature learning of user and item data, resulting in low accuracy. In this regard, a multi-convolutional neural network recommendation method is proposed, and the recommendation accuracy is significantly improved.

3. Recommendation model incorporating deep learning

In order to better extract the characteristics of users and movies based on traditional recommendation algorithms, this paper proposes a collaborative filtering recommendation model fused with convolutional neural networks. At the same time, Liu Chaohui et al. proposed a collaborative filtering recommendation model fused with time weight in [11], which makes the calculation of similarity more accurate. To this end, we introduce a time weighting factor into the model to consider the impact of ratings in different time periods on finding similar users. The improved model is shown in Figure 3.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

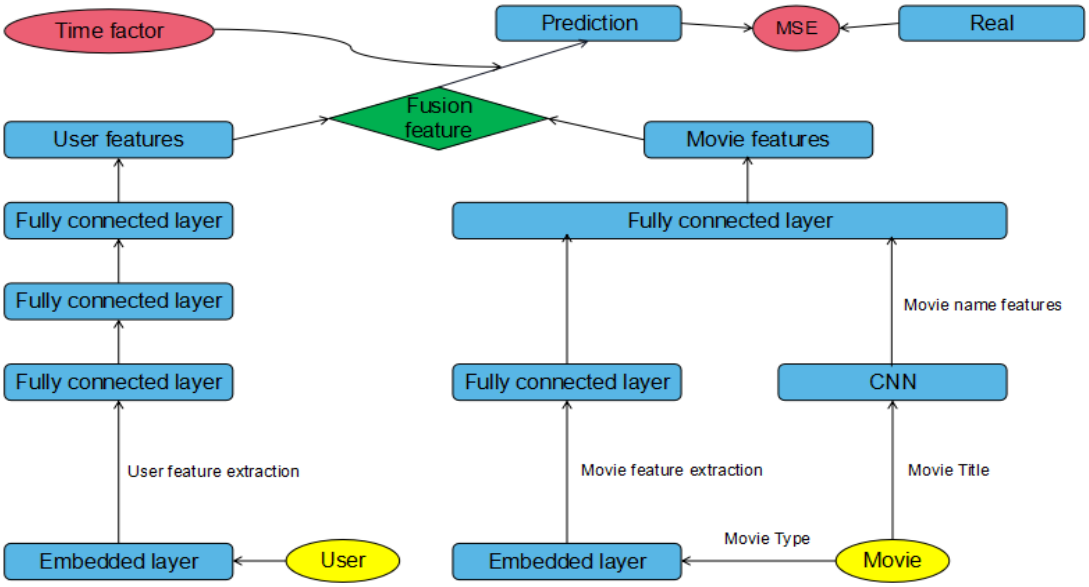

Figure 3. Recommendation model incorporating deep learning

3.1 Build the embedding layer

The embedding layer is mainly divided into the user embedding layer and the movie embedding layer. The function of the embedding layer is to input the description of users and movies in the data set into the model in the form of vectors for computer analysis. Usually processing some category fields to convert these fields into one-hot encoding. In the processing of movie names, Word2Vec is used to train word vectors.

3.2 Fully connected layer

Fully connected layers (FC) play the role of "classifier" in the entire convolutional neural network. If operations such as the convolutional layer, pooling layer, and activation function layer map the original data to the hidden layer feature space, the fully connected layer plays the role of mapping the learned "distributed feature representation" to the sample label space. Each node of the fully connected layer is connected to all the nodes of the previous layer, and is used to integrate the features extracted from the front. Due to its fully connected characteristics, generally the parameters of the fully connected layer are also the most. The forward full connection is calculated as follows:

$$\begin{aligned} a_1 &= W_{11} * x_1 + W_{12} * x_2 + \ldots + W_{1n} * x_n \\ a_2 &= W_{21} * x_1 + W_{22} * x_2 + \ldots + W_{2n} * x_n \\ &\ldots \\ a_n &= W_{n1} * x_1 + W_{n2} * x_2 + \ldots + W_{nn} * x_n \end{aligned} \tag{1}$$

Among them x_i (i=1,2,...,n) is the input part of the fully connected layer, a_i (i=1,2,...,n) is the output part of the fully connected layer, W is the weight coefficient, and b is Bias factor. When forward propagation is performed at the fully connected layer, W and b need to be updated.

3.3 Text convolutional neural network

For the features in the movie title, use convolutional neural network to extract. First, use the CBOW mode in Word2Vec to train the input movie name to obtain the feature vector:

$$T_i = (T_{i1}, T_{i2}, T_{i3}, \ldots, T_{in}) \in T^{k \times n} \tag{2}$$

Where T_{in} represents the feature vector of the nth character of the i-th movie, and k represents the number of movie names. After establishing the feature vector, start to build the convolution kernel, W_i represents the convolution kernel, its size is z, $W_i \in R^{Z \times Z}$, through W_i the convolution operation, you can get:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

$$m_i = f(T_{ij} \times W_i) \tag{3}$$

Where m_i represents the i-th The calculated value of the feature of each movie name vector. f is the non-linear activation function. This study uses the Relu function $R(x)$ as the activation function:

$$R(x) = max(0, x) \tag{4}$$

After the convolutional layer is established, the pooling layer is established. Assuming that feature $M_i = (m_1, m_2, m_3, \ldots, m_n)$ is the result obtained after j convolution operations, then M_i is the maximum pooling value, which is represented by q_i as:

$$q_i = max(M_i) = max\{m_1, m_2, m_3, \ldots, m_n\} \tag{5}$$

Finally, put q_i on the fully connected layer, and use the Relu activation function to calculate the feature vector of the movie name:

$$J_i = R(w_i q_i) \tag{6}$$

where w_i represents the weight of the fully connected layer of J_i.

3.4 Predictive score

After the user features and movie features are obtained through training, the two feature vectors are multiplied by vector, and the result is multiplied by the time weight to obtain a new prediction score. The time weight adopts the improved nonlinear forgetting function:

$$f(t) = e^{-\alpha \bullet [\frac{T_{final} - t}{T_{final} - T_{first}}]} \tag{7}$$

Where T_{final} is the user's last rating time, T_{first} is the user's first rating time, t represents the time when the user evaluates a certain movie at a certain moment, and can be obtained by calculation $f(t)$ the larger the user's evaluation time, the newer, The greater the impact on the recommended results.

4. Experiment procedure

In this paper, convolutional neural networks and nonlinear forgetting curve functions are applied to movie recommendation tasks, using Word2Vec pre-training word vectors. The model runs on Windows 10 system, the operating environment is Tensorflow 1.5.0, and the CPU is Intel(R) Core (TM) i5.

4.1 Experimental data

The data set used in this experiment is the standard data set MovieLens-1M, which contains the Users table, the Movies table and the Ratings table. The Users table contains UserID, Gender, Age, OccupationID and other fields. The table structure is shown in Table 1.

Table 1. Users table

	UserID	Gender	Age	Occupation
0	1	F	16	10
1	2	M	56	16
2	3	M	25	15

The Movies table contains three fields: MovieID, Title, and Genres. The table structure is shown in Table 2.

Table 2. Movies table

	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

The Ratings table contains four fields UserID, MovieID, Rating, and timestamps. The table structure is shown in Table 2.

Table 3. Ratings table

	UserID	MovieID	Rating	timestamps
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968

This data set contains 1 million ratings data from 6000 users on 4000 movies.

4.2 Experimental parameter settings

In this paper, the word vector length is set to 15, the text convolution sliding window is set to {2,3,4,5}, the number of convolution kernels is set to 8, and each batch is set to 258 samples for training, and the learning rate is set Is 0.0001, dropout is set to 0.5, and the number of iterations is set to 5.

4.3 Experimental results

This paper compares two experimental results, namely the collaborative filtering recommendation model fused with deep learning and the collaborative filtering recommendation model fused with deep learning and time factors. The loss regression curve of the collaborative filtering recommendation model fused with deep learning is shown in Figure 4.

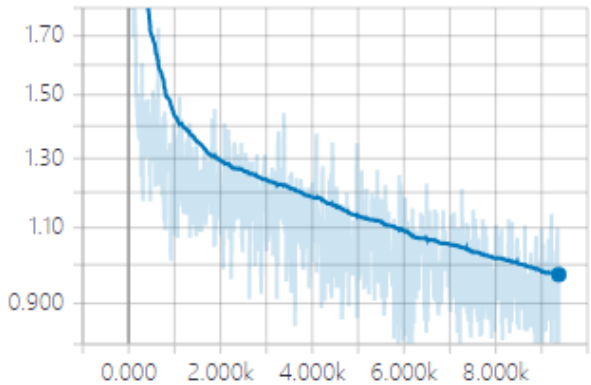

Figure 4. The loss regression curve of the collaborative filtering recommendation model fused with deep learning

The loss regression curve of the improved collaborative filtering recommendation model is shown in Figure 5.

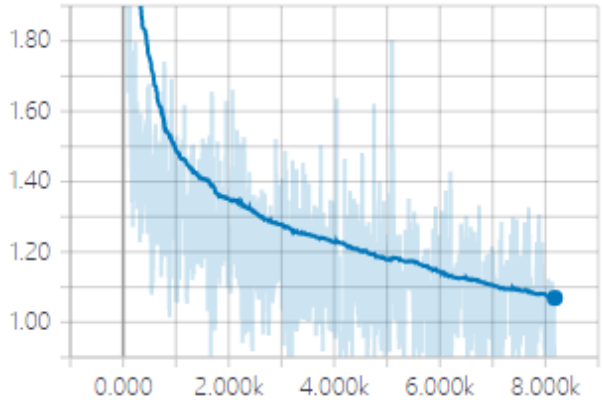

Figure 5. The loss regression curve of a collaborative filtering recommendation model that combines deep learning and time factors

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0043

After adding the time weighting factor to the original collaborative filtering recommendation model, the loss function is further reduced. After continuously adjusting the number of 20 iterations, it is found that when the number of iterations is 12, the recommended effect of the model achieves the optimal value. The index for judging the quality of the recommendation system is the absolute average error:

$$MAE = \frac{1}{n}\sum_{1}^{n} r_{ui} - \overset{\vee}{r}_{ui} \tag{8}$$

Where r_{ui} is the real rating of user u for movie i, and $\overset{\vee}{r}_{ui}$ is the predicted rating of user u for movie i. The obtained MAE value changes are shown in Table 4.

Table 4. Changes in MAE value

	N=10	N=20	N=30	N=40
DRM	0.85	0.83	0.82	0.84
T-DRM	0.83	0.81	0.80	0.82

It can be seen from the results of changes in the MAE value that when 30 similar users are selected, the recommendation effect is the best, and the MAE value is reduced after the time factor is introduced at the same time.

5. Conclusion

In most traditional recommendation systems, the ratings of movies are simply recommended to users through the calculation of recommendation algorithms, without considering the potential characteristics of users and movies. Incorporating deep learning into the traditional collaborative filtering recommendation algorithm can allow the model to automatically learn the features in the data, thereby improving the accuracy of the recommendation. After incorporating the time factor, it can solve the problem of inaccurate ratings caused by the long interval between user ratings, To make the score predicted by the model more credible.

This model does not consider the impact on recommendations after adding user comments. How to integrate user comments and the emotional features in user comments into the model is the next key task.

References

[1] W. Zhou, R. Li and W. Liu, "Collaborative Filtering Recommendation Algorithm based on Improved Similarity," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020, pp. 321-324, doi: 10.1109/ITOEC49072.2020.9141788.

[2] Y. Zeng, Y. Bi, J. Wang and Y. Lin, "Collaborative Filtering Recommendation Algorithm Optimization Based on User Attributes," 2015 8th International Symposium on Computational Intelligence and Design (ISCID), 2015, pp. 580-583, doi: 10.1109/ISCID.2015.91.

[3] R. Ji, Y. Tian and M. Ma, "Collaborative Filtering Recommendation Algorithm Based on User Characteristics," 2020 5th International Conference on Control, Robotics and Cybernetics (CRC), 2020, pp. 56-60, doi: 10.1109/CRC51253.2020.9253466.

[4] Y. Xiao and Q. Shi, "Research and Implementation of Hybrid Recommendation Algorithm Based on Collaborative Filtering and Word2Vec," 2015 8th International Symposium on Computational Intelligence and Design (ISCID), 2015, pp. 172-175, doi: 10.1109/ISCID.2015.211.

[5] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

[6] HUANG Li-Wei, LIU Yan-Bo, LI De-Yi, Deep Learning Based Recommender Systems, 2017, Vol.40, Online Publishing No.156

[7] Kim Y. Convolutional Neural Networks for Sentence Classification [J]. Eprint Arxiv, 2014

[8] H. Zarzour, B. Al shboul, M. Al-Ayyoub and Y. Jararweh, "A convolutional neural network-based reviews classification method for explainable recommendations," 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), 2020, pp. 1-5, doi: 10.1109/ SNAMS52053. 2020.9336529.

[9] Y. Zhang, J. Wan, A. Yang, C. Ji and J. Chu, "Research on Recommendation Method Based on Multi-Convolution Neural Network," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), 2019, pp. 420-424, doi: 10.1109/CISCE.2019.00099.

[10] M. Jianjun, "Research on collaborative filtering recommendation algorithm based on user behavior characteristics," 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), 2020, pp. 425-428, doi: 10.1109/ICBASE51474.2020.00096.

[11] C. Liu, C. Han, T. Chen and X. Kong. (2020) “Collaborative filtering recommendation algorithm based on penalty factors and time weights,”. Information Technology and Network Security, 17-21. doi:10.19358/ j.issn. 2096-5133.2020.05.004.